

VŠB – Technická univerzita Ostrava
Fakulta elektroniky a informatiky

BAKALÁŘSKÁ PRÁCE

VŠB – Technická univerzita Ostrava
Fakulta elektroniky a informatiky
Katedra měřicí a řídicí techniky

Návrh a realizace PID regulátoru na fyzikálním modelu
kuličky na tyči pomocí kompaktního
programovatelného automatu WinPac-8000

Design and Implementation of PID Controller on
Ball&Beam Model using Programmable Compact
Controller WinPac-8000

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 6. 5. 2011

.....

Poděkování

Touto cestou bych chtěl poděkovat panu Ing. Štěpánu Ožanovi, Ph.D. za konzultace, cenné rady a připomínky, které mi poskytl při řešení mé bakalářské práce. Dále bych chtěl poděkovat své rodině za podporu a důvěru při studiu na vysoké škole.

Abstrakt

Tato bakalářská práce se zabývá problematikou regulace pozice kuličky na tyči prostřednictvím řídicího systému PLC. Popisuje princip základních metod syntézy zpětnovazebních regulačních obvodů. Dále popisuje matematický a fyzikální model soustavy kuličky na tyči a její linearizaci v pracovním bodě.

Pro identifikovanou soustavu modelu je navržen PID regulátor a simulace regulačního obvodu v prostředí REX + Simulink. Následuje realizace navrženého algoritmu na programovatelném automatu WinPAC-8000 s využitím řídicího systému REX.

Jako graficko uživatelské rozhraní GUI je vytvořena vizualizace ve vybraném vizualizačním systému SCADA/HMI.

Klíčová slova

regulace, kulička na tyči, řídicí systém, PLC, PID regulátor, WinPAC-8000, REX, Simulink, GUI, vizualizace, SCADA/HMI

Abstract

This bachelor work is dealing with issues of regulation of position Ball&Beam trough the control system PLC. Work is describing primary methods of synthesis of feedback control loops. Further the the work is describing mathematics and physics model of system of Ball&Beam and her linearization of operating point.

For identified system of model is designed PID controller and simulation of control loop in surrounding REX + Simulink. Next is realization of designed algorithm on programmable automatic machine WinPAC-8000 with use of control system REX.

As graphical-user interface GUI is created visualization in selected visualization system SCADA/HMI.

Keywords

regulation, Ball&Beam Model, Control System, PLC, PID Controller, WinPAC-8000, REX, Simulink, GUI, Visualization, SCADA/HMI

Seznam použitých symbolů a zkratek

AP	přístupový bod (wifi)
C/C++	programovací jazyk
CPU	procesor
GUI	graficko – uživatelské rozhraní
I/O	vstup/výstup
IP	inteligentní protokol (IP adresa)
ISA	standardní matematický tvar regulátoru
K_d	derivační regulační konstanta
K_i	integrační regulační konstanta
K_p	proporcionální regulační konstanta
OPC	komunikační server
OS	operační systém
PAC	programovatelný automat
PD	proporcionálně – derivační regulátor
PI	proporcionálně – integrační regulátor
PID	proporcionálně – integrační – derivační regulátor
PIDU	blok PID regulátoru
PLC	programovatelný automat
RS-232	komunikační rozhraní
SCADA/HMI	vizualizační systémy
TCP/IP	komunikační protokoly
USB	univerzální port
VGA	standardní grafický adaptér

Obsah

1	Úvod.....	- 1 -
2	Popis modelu.....	- 2 -
2.1	Fyzikální popis modelu	- 2 -
2.2	Matematický popis modelu	- 4 -
3	Popis řídicího systému REX +WinPAC-8000.....	- 8 -
3.1	Popis WinPAC-8000	- 8 -
3.2	Popis řídicího systému REX.....	- 10 -
3.3	Běh systému REX a diagnostika	- 11 -
4	Linearizace v pracovním bodě	- 12 -
4.1	Blokové schéma linearizace v pracovním bodě	- 15 -
5	Návrh PID regulátoru.....	- 16 -
5.1	Metody nastavování regulátorů	- 17 -
5.2	Návrh regulátoru prostřednictvím pidtune, pidtool	- 18 -
5.3	Návrh regulátoru metodou PID	- 22 -
5.4	Porovnání navržených regulátorů	- 24 -
6	Návrh a realizace regulačního algoritmu	- 26 -
6.1	Simulace realizovaného regulátoru	- 29 -
7	Graficko – uživatelského rozhraní	- 31 -
8	Zhodnocení, závěr.....	- 34 -
	Literatura	I
	Seznam příloh.....	Chyba! Záložka není definována.

1 Úvod

Tato bakalářská práce se zabývá především regulací v automatizaci. Regulace (latinsky regula) obecně znamená řídicí či usměrňující činnost, ať už prováděnou člověkem, nebo automatickým zařízením (regulátorem). Ten patří neodmyslitelně k hlavním prvkům automatizace v průmyslu.

Pro regulaci pozice kuličky na fyzikálním modelu je regulátor realizován programovatelným automatem. Pro správnou funkci regulátoru je třeba popsat princip základních metod syntéz zpětnovazebních regulačních obvodů. Dále popsat matematický a fyzikální model soustavy kuličky na tyči. Vytvořit návrh PID regulátoru a realizovat jej.

Fyzikální model (soustava) je realizován ramenem, jehož ovládá servomechanismus, které vychyluje horizontální rovinu podlaží kuličky a tím mění její polohu. Poloha kuličky je snímána čidlem a informace o poloze je tvořena hodnotou napětí v definovaném rozsahu (výstup soustavy). Servomotor je rovněž řízený hodnotou napětí v definovaném rozsahu (vstup soustavy). Vstup i výstup soustavy je připojen přes komunikační rozhraní na řídicí systém.

Řídicí systém je realizován programovatelným automatem WinPAC-8000 s využitím řídicího systému REX. Navržený PID regulátor je ve vývojovém prostředí RexDraw realizován a aplikován do regulačního schématu. V tomto schématu je soustava odkazována jejím vstupem a výstupem. Další bloky schématu slouží jako přepočty regulační odchylky na stupně výkyvu ramene a ke kalibraci vychylování.

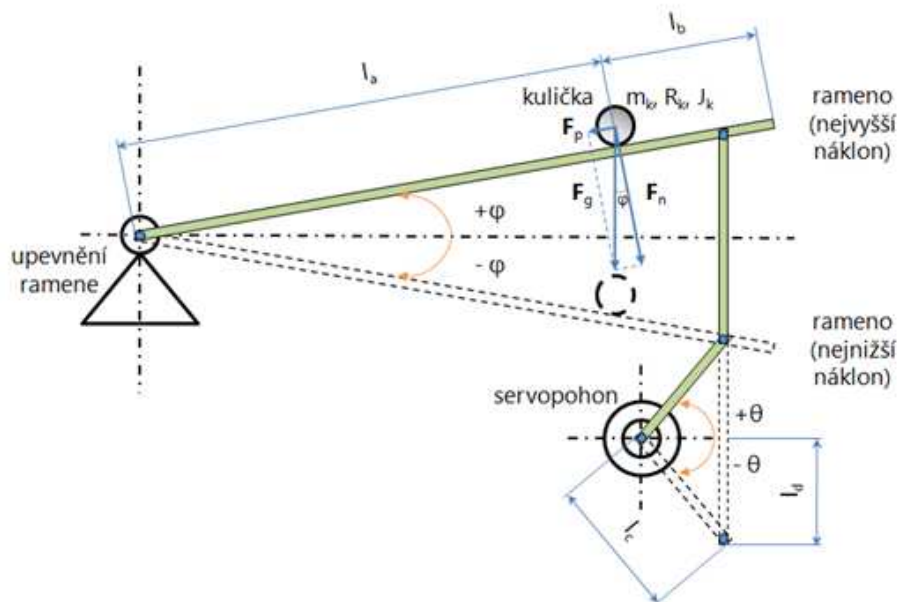
Navržený regulační obvod je simulován v simulačním prostředí Matlab Simulink a v simulačním prostředí REX.

Pro uživatelské rozhraní s regulátorem je vytvořeno grafické uživatelské rozhraní GUI. Toto rozhraní je vytvořeno ve vizualizačním systému SCADA/HMI.

2 Popis modelu

Jedná se o fyzické zpracování modelu kuličky na tyči. Hlavní úlohu modelu hraje ocelová kulička. Ta je umístěna na kolejničce. Kolejničku tvoří dvě rovnoběžně napnutá ocelová lanka (resp. tyč), která jsou na jednom konci upevněná ve statickém bodě. Na druhém konci jsou lanka upevněná na tyč, kterou ve vertikální poloze vychyluje nahoru a dolů servomechanismus. Pootočením osy servomotoru, se mění poloha vychylovací tyče a tím se také mění náklon roviny kolejničky. Při změně úhlu, resp. náklonu roviny kolejničky, se vlivem gravitační síly kulička pohybuje. (viz. Obr. 1) Pozn. popis výukového modelu kuličky na tyči uvedený v této kapitole je převzat z literatury [8].

2.1 Fyzikální popis modelu



Obr. 1 Schéma fyzikálního modelu kuličky na tyči

Na **Obr. 1** lze vidět fyzikální schéma modelu. Délka ramene je definována součtem vzdáleností l_a a l_b , kde l_b je požadovaná pozice kuličky. Kulička je definovaná hmotným středem, jenž má hmotnost m_k a moment setrvačnosti J_k . Poloměr kuličky je označen R_k . Při změně úhlu θ osy servopohonu se pootočí vychylovací tyč, která je připevněna kloubem k rameni modelu. Tímto se rameno naklání o úhel φ . Výška zdvihu opory při otáčení servopohonu je označena vzdáleností l_d a poloměr osy otáčení servopohonu je označen jako vzdálenost l_c . Při naklonění ramene v úhlu $\varphi = 0^\circ$ působí na kuličku gravitační síla F_g . Na tuto sílu působí vlivem podlaží v opačném směru normálová síla F_n . Tyto síly se navzájem odečítají a jejich nulová výslednice ponechává kuličku v klidu. Pokud se

rameno nakloní o úhel $\varphi \neq 0^\circ$, posmění se směř vektoru síly F_n o tentýž úhel. Rozvorem vektorů sil F_g a F_n vzniká výsledný vektor potenciální síly F_p , která pohybuje kuličkou.

Model lze rozdělit do tří důležitých částí:

- **mechanická část** – Mech. část se obsahuje všechny mechanické komponenty jako např. ocelovou kuličku, kolejnici z ocelových lanek, rameno se statickým upevněním na jednom konci, pomocnou zdvihadí oporu upevněnou na hřideli servopohonu a kovový kryt modelu.
- **elektrickou část** – El. část zahrnuje veškerou řidící elektroniku na modelu. Tj. analogový vstup a výstup pro řidící systém, obvody pro vyhodnocování polohy ramene, obvody pro vyhodnocování polohy kuličky, obvod řízení celého modelu, indikace chodu modelu, napájecí zdroj pro celý model s přístrojovou svorkovnicí a komunikační rozhraní.
- **programová část** – Zahrnuje řidící systém WinPAC-8000 s regulačním programem vytvořeným ve vývojovém prostředí RexDraw a vizualizačním prostředím.

Model (resp. zdroj modelu) je napájen síťovým napájením 230 [V_{AC}]. Součástí napájecího modulu je tavná pojistka. Vypínač je realizován páčkovým spínačem. Zelená led dioda indikuje zapnutý stav. Vstupy a výstupy jsou realizované komunikačním rozhraním RS232 s devíti pinovým konektorem CANON9. Tyto analogové vstupy a výstupy pracují v napěťových rozsazích 0 [V_{DC}] až 10 [V_{DC}]. Napětí analogového vstupu je akční zásah pro servopohon. Hodnota napětí analogového výstupu reprezentuje pozici kuličky.

Hodnoty jednotlivých fyzikálních veličin popisujících ocelovou kuličku, polohu kuličky, polohu ramene a pomocnou zdvihadí oporu se servopohodem se nachází v tabulce.

(viz. **Tab. 1** a **Tab. 2**)

Ocelová kulička			
Označení veličiny	Hodnota	Jednotka	Význam
m_k	$m_k = 0,056$	[kg]	hmotnost kuličky
R_k	$R_k = 0,012$	[m]	poloměr kuličky
J_k	$J_k = \frac{2}{5} \cdot m_k \cdot R_k^2 = 3,2256 \cdot 10^{-6}$	[kg · m ²]	moment setrvačnosti kuličky
V_k	$V_k = \frac{4}{3} \cdot \pi \cdot R_k^3 = 7,2382 \cdot 10^{-6}$	[m ³]	objem kuličky
ρ_k	$\rho_k = 7700$	[kg · m ⁻³]	hustota kuličky

Tab. 1 Hodnoty fyzikálních veličin kuličky

Poloha kuličky			
Označení veličiny	Hodnota	Jednotka	Význam
l	$l = 0,95$	$[m]$	aktivní délka ramene
l_b	$l_b = l - l_a$	$[m]$	požadovaná vzdálenost kuličky
l_a	$l_a = l - l_b$	$[m]$	vzdálenost středu kuličky od pevného konce ramene
g	$g = 9,80665$	$[m \cdot s^{-2}]$	tižové zrychlení normální
$F_g = G$	$F_g = m_k \cdot g = 0,550$	$[N]$	tižová síla
F_p	$F_p = F_g \cdot \sin(\varphi)$	$[N]$	potenciální síla kuličky
F_n	$F_n = F_g \cdot \cos(\varphi) + m \cdot \ddot{\varphi} \cdot l_b$	$[N]$	normálová síla kuličky
Poloha ramene			
Označení veličiny	Hodnota	Jednotka	Význam
φ	$\varphi \in \langle -0,01; +0,01 \rangle$	$[rad]$	interval úhlu sklonu ramene
Pomocná zdvihací opora se servopohonem			
Označení veličiny	Hodnota	Jednotka	Význam
l_c	$l_c = 0,019$	$[m]$	poloměr osy otáčení servopohonu
l_d	$l_d = \varphi \cdot l = 0,095$	$[m]$	délka zdvihu opory při otáčení servopohonu nahoru nebo dolů
θ	$\theta \in \langle -\frac{\pi}{6}; +\frac{\pi}{6} \rangle$	$[rad]$	interval úhlu natočení servopohonu

Tab. 2 Hodnoty jednotlivých fyzikálních veličin popisujících polohu kuličky, polohu ramene a pomocnou zdvihací oporu se servopohonem

2. 2 Matematický popis modelu

Základ matematického popisu modelu tvoří lagrangeovské pojetí mechaniky ve formě Lagrangeových rovnic II. druhu, umožňujících vytvoření pohybových rovnic soustavy hmotných bodů zavedením tzv. zobecněných souřadnic. [2]

Nejprve je nutno nalézt matematické relace pro kinetickou energii E_k a potenciální energii E_p soustavy kuličky na tyči. Ve výsledné kinetické energii soustavy uvažujeme čtyři druhy kinetické energie – dva druhy pro kuličku (translace, rotace) a dva druhy pro rameno, respektive tyč.

Pro kuličku platí následující relace dle [3]:

translační energie kuličky a rotační energie kuličky

$$E_{k1} = \frac{1}{2} m_k \dot{x}^2(t) + \frac{1}{2} \cdot \frac{J_k}{R_k^2} x^2(t) = \frac{1}{2} \dot{x}^2(t) \left[m_k + \frac{J_k}{R_k^2} \right] \quad (1)$$

Pro tyč bez kuličky a s kuličkou platí následující relace, tedy:

kinetická energie tyče a kinetická energie tyče s kuličkou

$$E_{k2} = \frac{1}{2} \cdot J_t \cdot \dot{\varphi}^2(t) + \frac{1}{2} \cdot m_k \cdot x^2(t) \cdot \dot{\varphi}^2(t) = \frac{1}{2} \cdot \dot{\varphi}^2(t) \cdot [J_t + m_k \cdot x^2(t)] \quad (2)$$

Výsledná kinetická energie soustavy E_k je dána součtem kinetických energií kuličky (1) a kinetických energií tyče bez kuličky a s kuličkou (2), tedy:

$$E_k = E_{k1} + E_{k2} = \frac{1}{2} \cdot \dot{x}^2(t) \cdot \left[m_k + \frac{J_k}{R_k^2} \right] + \frac{1}{2} \cdot \dot{\varphi}^2(t) \cdot [J_t + m_k \cdot x^2(t)] \quad (3)$$

<i>kde</i>	$x(t) \equiv l_b(t)$	okamžitá poloha kuličky na tyči	$[m]$
	$\dot{x}(t) \equiv \dot{l}_b(t)$	okamžitá rychlost kuličky na tyči	$[m \cdot s^{-1}]$
	$\varphi(t)$	okamžitý úhel (fáze) sklonu tyče	$[rad]$
	$\dot{\varphi}(t)$	okamžitý úhlový kmitočet tyče	$[rad \cdot s^{-1}]$
	J_t	moment setrvačnosti tyče (ramene)	$[kg \cdot m^2]$

Ve výsledné potenciální energii soustavy uvažujeme potenciální sílu kuličky F_p pohybující se po nakloněné rovině ve formě nakloněné tyče (viz **obr. 1**), přičemž platí následující relace, tedy:

$$F_p = F_g \sin(\varphi) = G \sin(\varphi) = m_k g \sin(\varphi) \quad (4)$$

Výsledná potenciální energie soustavy E_p je dána relací, tedy:

$$E_p = F_p x(t) = m_k g x(t) \sin(\varphi) \quad (5)$$

V dalším kroku je nutno určit práci nekonzervativních sil $W(t) \equiv W$, tj. Lagrangeova rovnice II. druhu (v Eulerově-Lagrangeově tvaru) nebude homogenní čili rovna nule a bude reprezentována tzv. zobecněnou potenciálovou funkcí $U[\dot{x}(t), x(t), t]$, respektive $U[\dot{\varphi}(t), \varphi(t), t]$, přičemž zobecněnými souřadnicemi jsou okamžitá poloha kuličky $x(t)$ a okamžitý úhel sklonu tyče $\varphi(t)$. Příslušné parciální derivace jsou dány relacemi, tedy:

parciální derivace práce dle polohy kuličky

$$\frac{\partial}{\partial x(t)} \{W(t)\} = -b\dot{x}(t) - \frac{\xi}{R_k} F_n \text{sign}[\dot{x}(t)] \quad (6)$$

parciální derivace práce dle úhlu sklonu tyče

$$\frac{\partial}{\partial \varphi(t)} \{W(t)\} = F_s l \cos(\varphi) \quad (7)$$

<i>kde</i>	b	odpor prostředí při pohybu kuličky	$[kg \cdot s^{-1}]$
	ξ	rameno valivého odporu	$[m]$
	F_s	síla převodu servopohonu	$[N]$

Nyní sestavíme obecný předpis k rovnicím (6) a (7) levé strany ve tvaru Lagrangeovy rovnice

II. druhu pro nekonzervativní systém, tedy:

$$\frac{d}{dt} \left\{ \frac{\partial}{\partial \dot{q}_i(t)} L(t) \right\} - \frac{\partial}{\partial q_i(t)} \{L(t)\} = \frac{\partial}{\partial q_i} \{W(t)\} \quad (8)$$

<i>kde</i>	$L(t) \equiv L = E_k - E_p$	lagrangián	$[J]$
	$q_i = 1(t) = x(t)$	okamžitá poloha kuličky	$[m]$
	$\dot{q}_i = 1(t) = \dot{x}(t)$	okamžitá rychlost kuličky na tyči	$[m \cdot s^{-1}]$
	$q_i = 2(t) = \varphi(t)$	okamžitý úhel (fáze) sklonu tyče	$[rad]$
	$\dot{q}_i = 2(t) = \dot{\varphi}(t)$	okamžitý úhlový kmitočet tyče	$[rad \cdot s^{-1}]$

Po proderivování a dosazení do (8) dostáváme následující relace, tedy:

pro okamžitou polohu kuličky – viz (6)

$$\frac{\partial}{\partial x(t)} \{W(t)\} = \left[m_k + \frac{J_k}{R_k^2} \right] \ddot{x}(t) - m_k x(t) \dot{\varphi}^2(t) + m_k g \sin(\varphi) \quad (9)$$

pro okamžitý úhel sklonu tyče – viz (7)

$$\frac{\partial}{\partial \varphi(t)} \{W(t)\} = [J_t + m_k x^2(t)] \ddot{\varphi}(t) + m_k g x(t) \cos(\varphi) \quad (10)$$

při lagrangiánu

$$L = E_k - E_p = \frac{1}{2} \dot{x}^2(t) \left[m_k + \frac{J_k}{R_k^2} \right] + \frac{1}{2} \dot{\varphi}^2(t) [J_t + m_k x^2(t)] - m_k g x(t) \sin(\varphi) \quad (11)$$

Položením rovností mezi relace (6) a (9) a relace (7) a (10) lze získat následující relace:

$$-b\dot{x}(t) - \frac{\xi}{R_k} F_n \text{sign}[\dot{x}(t)] = \left[m_k + \frac{J_k}{R_k^2} \right] \ddot{x}(t) - m_k x(t) \dot{\varphi}^2(t) + m_k g \sin(\varphi) \quad (12)$$

$$F_s l \cos(\varphi) = [J_t + m_k x^2(t)] \ddot{\varphi}(t) + m_k g x(t) \cos(\varphi) \quad (13)$$

Relace (12) vyjadřuje vliv náklonu na pohyb kuličky, relace (13) pak vyjadřuje vliv kuličky na náklon tyče, přičemž tuto relaci zanedbáváme. Pro odvození přenosové funkce této soustavy (vztah mezi náklonem tyče a pozicí kuličky) budeme dále uvažovat relaci (12), kterou lze zredukovat zanedbáním ramena valivého odporu a odstředivé síly kuličky a zlinearizovat okolo následujícího ekvilibria:

$$\varphi(t) = \dot{\varphi}(t) = x(t) = \dot{x}(t) = 0 \quad (14)$$

Při linearizaci dále uvažujeme:

$$\sin(\varphi) \cong \varphi \quad \varphi \in \left\langle 0; \frac{\pi}{36} \right\rangle [rad] \quad (15)$$

uvažování relací (14) a (15) lze relaci (12) přepsat do tvaru homogenní OLDR 2. řádu:

$$\left[m_k + \frac{J_k}{R_k^2} \right] \ddot{x}(t) + b\dot{x}(t) + m_k g \varphi = 0 \quad (16)$$

Přenosová funkce (vnější popis) v Laplaceově transformaci při nulových počátečních podmínkách a volbou nulového odporu prostředí je dána pak dána relací:

$$\begin{aligned} \hat{G}_s(s) &= \frac{\hat{L}\{x(t)\}}{\hat{L}\{\varphi(t)\}} = \frac{\hat{X}(s)}{\hat{\varphi}(s)} = \frac{-m_k g}{\left[m_k + \frac{J_k}{R_k^2} \right] s^2 + bs} = \frac{-F_g}{\left[m_k + \frac{J_k}{R_k^2} \right] s^2 + bs} = \frac{b_0}{a_2 s^2 + a_1 s} = \\ &= \frac{-0,550}{0,077 s^2} = \frac{-7,1428}{s^2} \end{aligned} \quad (17)$$

Stavový popis (vnitřní popis) lze získat substitucí lineární diferenciální rovnice 2. řádu v relaci (16) na soustavu dvou lineárních diferenciálních rovnic 1. řádu:

$$\dot{x}_1(t) = \frac{1}{\left[m_k + \frac{J_k}{R_k^2} \right]} [-bx_1(t) - m_k g u(t)] \quad (18a)$$

$$\dot{x}_2(t) = x_1(t) \quad (18b)$$

při substituci

$$x_1(t) = \dot{x}(t); x_2(t) = x(t); u(t) = \varphi(t) \quad (19)$$

Dle očekávání jsme získali astatický systém 2. řádu. Vzhledem k původním rovnicím (12) a (13) lze očekávat nelineární chování kuličky na krajích nakláněné tyče, kde výrazněji působí zanedbaná odstředivá síla. Další významnou nelinearitou je samozřejmě konečná délka tyče a v praxi se projeví i nerovnosti vodících kolejniček. [2]

3 Popis řídicího systému REX +WinPAC-8000

Řídicí systém je realizován programovatelným automatem WinPAC-8000. Ten je připojen k soustavě (fyzikální model kuličky na tyči) přes komunikační rozhraní RS-232. V programovatelném automatu běží operační systém Windows CE 5.NET. Software řídicího systému je vytvořen ve vývojovém prostředí RexDraw a importován do automatu přes komunikační rozhraní Ethernet (wifi).

3.1 Popis WinPAC-8000

WinPAC-8000 je druhá generace PAC od ICP DAS. Je vybaven CPU (520MHz) a operačním systémem Windows CE.NET 5.0. Tento automat disponuje konektivitou VGA, USB, Ethernet, RS232/485 a sloty pro vysoce výkonné moduly paralelních a sériových vstupů/výstupů.

Operační systém automatu Windows CE 5.0 má mnoho výhod včetně hard real-time módu. Má malou velikost jádra, rychlé zpracování přerušení na hlubší úrovni, dosažitelné deterministické ovládání a nízké náklady. Použití systému Windows 5.0 v CE.NET dává WinPAC-8000 možnost spustit software založený na PC jako je Visual Basic.NET, Visual C#, Embedded Visual C++, SCADA software, Soft PLC.



Obr. 2 WinPAC-8000

- **Výkonný CPU modul** – PXA270 nebo kompaktní procesor (32-bit a 520MHz), 128 MB SDRAM, 31 MB vestavěný Flash Disk
- **Vestavěný VGA port** – Může být připojen přímo k LCD displeji. Uživatelé mohou Pracovat v HMI nebo SCADA software (běžící na WinPAC-8000) s displejem, klávesnicí a myší, stejně jako na PC.
- **64-bitové hardwarové číslo** – Každé pořadové číslo WinPAC-8000 je odlišné. Uživatelé mohou přidat kontrolní mechanismus k jejich AP, aby se zabránilo připojení neoprávněným uživatelům.

- **I/O modul Hot Swap** – WinPAC-8000 nabízí hot swap, což znamená, že není nutné vypínat PAC pro výměnu modulů. OS poskytuje funkce zasílání plug-in a odstraňování zpráv uživatelských aplikací. Pomocí této funkce mohou uživatelé navrhovat své vlastní plug-and-play aplikace.



Obr. 3 I/O modul Hot Swap

- **Bohaté I/O rozšíření** – Vedle lokálních I/O slotů je WinPAC-8000 vybaven několika porty RS-232/485, také porty pro sériové připojení I/O a Ethernet I/O. Dále CAN a FRnet komunikační modul v místním slotu.

- **Built-in Flash Disk 31 MB** – Uživatelé mohou ukládat svá data na kartu Micro SD nebo na USB Flash disk. Pro některé vibrační prostředí je pak vestavěný Flash disk jako nejvhodnější volba paměťového média.

- **Dual-Backup baterie SRAM(512kB)** – Pro uložení důležitých dat při vypnutí napájení, je ideálním řešením. WinPAC je vybaven 512 SRAM se dvěma bateriemi Li, které udržují data při vypnutí. Obě baterie mohou průběžně napájet paměť po dobu až 5 let. Dvojitá napájení paměti může zabránit ztrátě dat při výměně nové baterie.

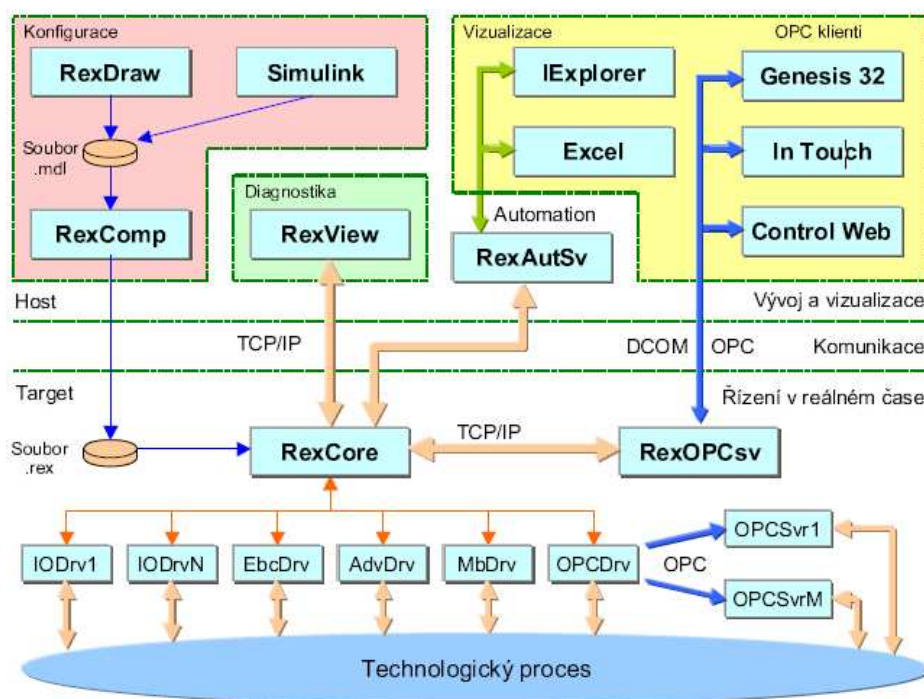
- **Dva Ethernet porty** – WinPAC-8000 nabízí dva ethernetové porty. Tyto porty mohou být použity k provedení redundantní ethernetové komunikace a samostatné komunikace Ethernet (jedno pro globální internet, druhý pro soukromý ethernet).

- **Redundantní napájecí vstupy** – Tohle řešení napájení je výhodné v případě, že dojde k výpadku proudu zdroje. Poté se přepne na sekundární napájecí vstup, aby se nezabránilo nepřetržitému provozu.

- **Ventilace** – Každý WinPAC-8000 je umístěn v plastové krabici se sloupcovým profilem a ventilátorem. Ten může pomocí chlazení uvnitř boxu umožnit provoz PAC v okolní teplotě mezi -25°C a 75°C.

3. 2 Popis řídicího systému REX

Řídicí systém REX je otevřený a škálovatelný systém vhodný pro vnořené (embedded) řízení, přenositelný na různé platformy s překladači jazyka C a C++ od jednoúčelových řídicích desek s jednoduchou exekutivou reálného času až po procesní stanice se standardními operačními systémy (Windows NT, 2000, XP, VxWorks, apod.). Kompatibilita řídicího systému REX s programovým balíkem Simulink byla jednou ze základních myšlenek návratu systému REX[5].



Obr. 4 Struktura řídicího systému REX

Na obrázku 4 je zobrazena struktura řídicího systému REX. V horní polovině lze vidět bloky pro vizualizaci, operátorského ovládání a komponenty vývojového prostředí. Všechny tyto vývojové nástroje jsou určeny pro operační systém Windows 95/98/ME/NT/2000/XP. Pro vizualizaci jsou zobrazeny standardní vizualizační nástroje. Nástroj pro připojení vizualizace se systémem jsou buď OPC, nebo DOM a DCOM.

Spodní část obrázku naznačuje strukturu cílového prostředí (Target), které realizuje vlastní řízení v reálném čase[5].

Spojovacím článkem mezi prostředím Host a Target je komunikační vrstva (viz Obr. 4). Nejčastěji používaným protokolem pro komunikaci je standard TCP/IP, nad nímž je vybudován vlastní diagnostický protokol systému REX[5].

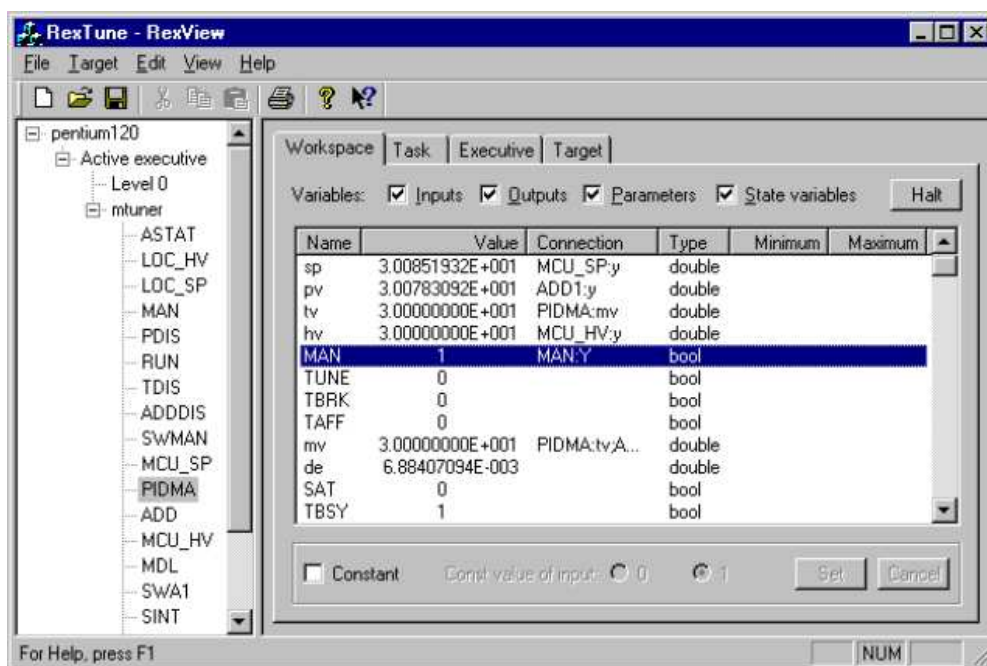
3.3 Běh systému REX a diagnostika

Jádro řídicího systému REX tvoří program RexCore, který je trvale provozovaný na cílové platformě (WinPAC-8000). Pro možnosti pozorování řídicího systému zejména v procesu uvádění do provozu jsou třeba ještě další nástroje. Kromě zmíněných vizualizačních systémů slouží k detailnímu pozorování program RexView.

Jádro řídicího systému RexCore je dost složitý program provádějící paralelně různé činnosti obvyklé v řídicích systémech[5]. Tzn. zpracovává různé subsystemy jako např. subsystem reálného času, vstupně-výstupní subsystem, algoritmický subsystem, diagnostický subsystem a archivační subsystem. Jednotlivé subsystemy jsou prováděny v pořadí dle priority.

Program RexView umožňuje sledovat, co se děje v jádře řídicího systému REX při jeho běhu, a proto je velmi důležitým nástrojem při uvádění řídicího systému do provozu i v případě vzniku nějakých problémů již během rutinního provozu[5]. V programu lze pozorovat detailní informace o všech subsystemech jádra. Pomocí komunikace TCP/IP protokolu je snadné připojení k běžícímu jádru do lokální počítačové sítě i do vzdálené sítě jako je například Internet.

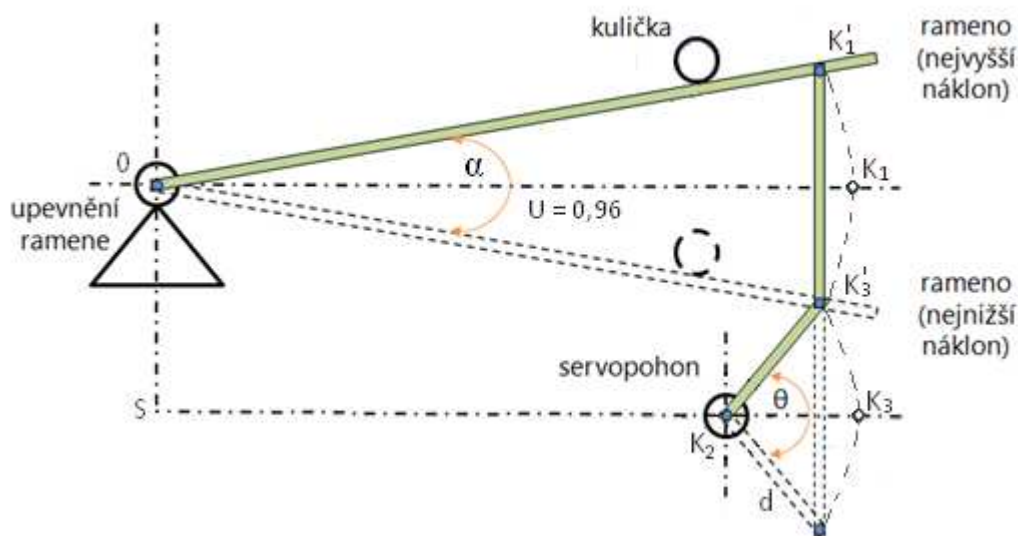
Na obrázku 5 je sejmuta ukázka obrazovky programu RexView na demonstračním příkladu. Podrobným popisem programu RexView na konkrétním příkladě tohoto tématu se bude zabývat jedna z následujících kapitol.



Obr. 5 Ukázka obrazovky programu RexView

4 Linearizace v pracovním bodě

Linearizace v pracovním bodě vychází z matematicko-fyzikálního modelu kuličky na tyči. Jak lze pozorovat z obrázku 6, pro správné určení pracovního bodu využíváme polohy ramene označené bodem K_x a bodem K'_x . Dále vycházíme z trojúhelníků tvořených z úhlů α , θ a příslušné délky ramene U a d .



Obr. 6 Obrázek pro linearizaci v pracovním bodě

Podle výše zmíněné relace se vychází z trojúhelníků:

odvození úhlu α

$$\sin \alpha = \frac{\Delta_{vert}}{U} \Rightarrow \Delta_{vert} = U \cdot \sin \alpha \quad (20)$$

$$\sin \theta = \frac{\Delta_{vert}}{d} \Rightarrow \Delta_{vert} = d \cdot \sin \theta \quad (21)$$

Z těchto vztahu vyplývá, že pokud jsou si výšky vertikál rovny, tak:

rovnost výšek vertikál

$$d \cdot \sin \theta = U \cdot \sin \alpha \quad (22)$$

Za předpokladu, že:

$$\sin \alpha \approx \alpha \quad (23)$$

Je úhel α vyjádřený následovně:

odvozený úhel α

$$\alpha = \frac{d \cdot \sin \theta}{U} \quad (24)$$

<i>kde</i>	α	interval úhlu sklonu ramene	$[rad]$
	θ	interval natočení úhlu servomotoru	$[rad]$
	U	délka ramene	$[m]$
	d	poloměr osy otáčení servopohonu	$[m]$
	S	Výška osy servomotoru	$[m]$

Vychází se z parametrů, kde:

parametry modelu

$$d = 0,02$$

$$U = 0,96$$

$$S = -0,11$$

Odvození souřadnic bodů K_x :

souřadnice polohy bodu K_1

$$\begin{aligned} X_{K1} &= U \cdot \cos \alpha \\ Y_{K1} &= U \cdot \sin \alpha \end{aligned} \quad (25)$$

souřadnice polohy bodu K_2

$$\begin{aligned} X_{K2} &= U - d = 0,94 \\ Y_{K2} &= -0,11 = S \end{aligned} \quad (26)$$

souřadnice polohy bodu K_3

$$\begin{aligned} X_{K3} &= X_{K2} + d \cdot \cos \theta = 0,94 + 0,02 \cdot \cos \theta \\ Y_{K3} &= S + d \cdot \sin \theta = -0,11 + 0,02 \cdot \sin \theta \end{aligned} \quad (27)$$

Za předpokladu, že vzdálenost mezi bodem K_1 a K_2 je konstantní, postup k výpočtu linearizace pracovního bodu je následující:

vzdálenost (výška) na intervalu $\{K_1, K_2\}$

$$v\{K_1, K_2\} = konst = |S|$$

vyjádření S pomocí souřadnic bodů $\{K_1, K_2\}$

$$|S| = \sqrt{(Y_{K3} - Y_{K1})^2 + (X_{K3} - X_{K1})^2} \quad (28)$$

Dosazení do vzorce (28):

$$0,11 = \sqrt{(-0,11 + 0,02 \cdot \sin \theta - U \cdot \sin \alpha)^2 + (0,94 + 0,02 \cos \theta - U \cdot \cos \alpha)^2}$$

s předpokladem že

$$\begin{bmatrix} \sin \alpha \approx \alpha \\ \cos \alpha \approx 1 \end{bmatrix}$$

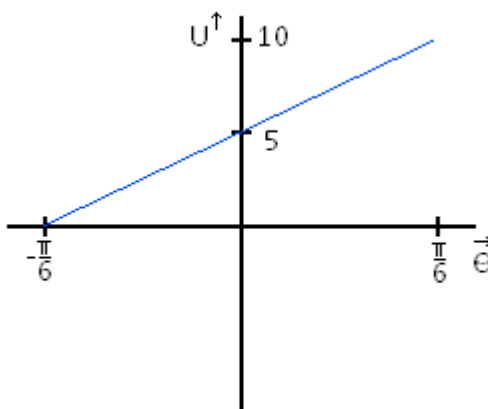
$$0,11 = \sqrt{(-0,11 + 0,02 \cdot \sin \theta - 0,96\alpha)^2 + (0,94 + 0,02 \cdot \cos \theta - 0,96)^2}$$

Po provedení výpočtu je výsledek:

$$\underline{\underline{\alpha = \frac{\sin \theta}{48} \Rightarrow \theta = \arcsin(48\alpha)}}$$

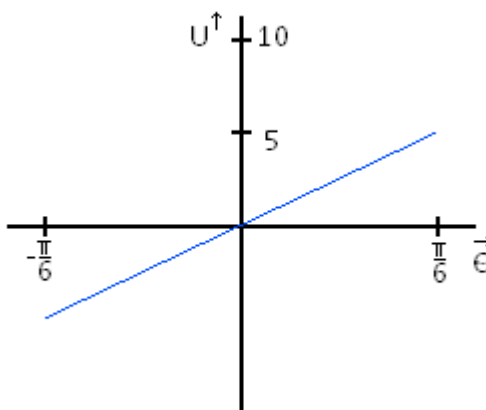
Provedení linearizace v grafu vypadá následovně:

$$\theta \in \left\langle -\frac{\pi}{6}; \frac{\pi}{6} \right\rangle$$



Obr. 7 Graf křivky pracovního bodu

$$\Delta U = U - U_0 = U - 5$$



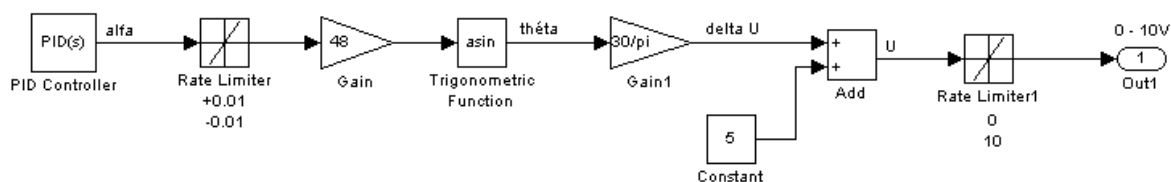
Obr. 8 Graf linearizace pracovního bodu

Výsledné ΔU je:

$$\Delta U = \frac{5}{\frac{\pi}{6}} \theta = \frac{30}{\pi} \cdot \theta \quad (29)$$

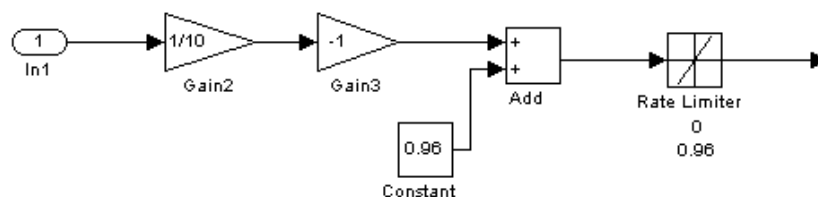
4.1 Blokové schéma linearizace v pracovním bodě

V předchozí kapitole byla teoreticky odvozena a vypočtena linearizace pracovního bodu. Pro praktické využití v programovacím prostředí RexDraw byly tyto výpočty modifikovány do blokového schématu.



Obr. 9 Blokové schéma linearizace v pracovním bodě,
přepočet náklonu α na výstupní napětí regulátoru

Na obrázku 9 je schématicky zobrazen výstup PID regulátoru náklonu ramene α , který po limitaci je podle teoretických výpočtů předchozí kapitoly vynásoben konst. 48 a následovně přepočítán fnc. asin. Výsledkem je úhel pootočení hřídele servopohonu θ . Vynásobením tohoto úhlu s konst. $30/\pi$ dosáhneme převodu na napětí výstupu regulátoru ke kterému je přičtena konst. 5 pro linearizaci pracovního bodu.



Obr. 10 Blokové schéma přepočtu vstupního napětí regulátoru na vzdálenost
polohy kuličky od počátečního bodu ramene

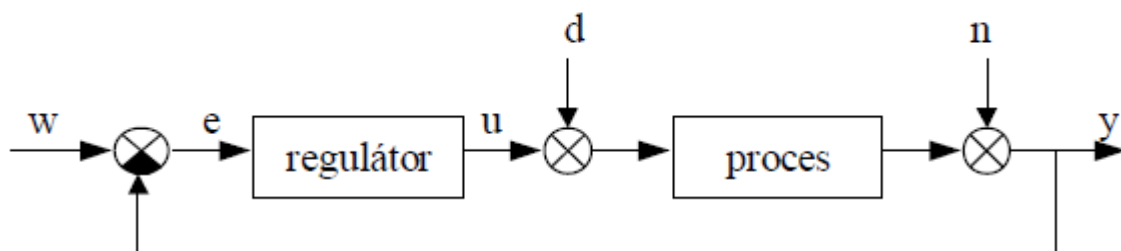
Na obrázku 10 je schématicky zobrazen vstup PID regulátoru, jenž přepočítává polohu kuličky (resp. napětí v rozsahu 0 – 10V) na hodnotu délky vzdálenosti kuličky od počátku ramene v měřítku na metr.

5 Návrh PID regulátoru

Proporcionálně – integračně – derivační (PID) regulátory jsou bezkonkurenčně nejpoužívanějšími regulátory v průmyslu. Uvádí se dokonce, že až 95% všech regulačních algoritmů je typu PID a že velká většina z nich navíc využívá pouze proporcionální a integrační složku[5].

Za počátek éry PID regulátorů (v podobě, jak je známe dnes) lze považovat období 1915-1940, kdy vznikaly proslulé regulační firmy Bristol, Fisher, Foxboro, Honeywell, Leeds & Nortrup a Taylor Instrument[5].

Schéma jednoduché regulační smyčky je na obr. 11. Regulátor a řízený proces jsou zde propojeny do uzavřené smyčky se zápornou zpětnou vazbou. Základní požadavek na její funkce lze vyjádřit následovně: požadujeme, aby regulovaná veličina y v každém okamžiku co možná nejpřesněji sledovala požadovanou hodnotu w a to nezávisle na působení poruchových veličin d a n a na změnách dynamických vlastností řízeného systému[5].



Obr. 11 Jednoduchá regulační smyčka [5]

Snaha zlepšit stabilitu uzavřené smyčky s PI regulátorem vedla k zavedení derivační (D) složky. Ideální PID regulátor (v lineární oblasti) realizuje tedy zákon řízení ve tvaru

$$u(t) = K \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (30)$$

kde parametr T_d se nazývá derivační časová konstanta. Složku PD v (30) můžeme chápat jako modifikovanou P složku, kde je odchylka $e(t)$ nahrazena predikovanou odchylkou $e(t + T_d)$, neboť z Taylorova rozvoje plyne

$$e(t + T_d) \doteq e(t + T_d) = e(t) + T_d \frac{de(t)}{dt} \quad (31)$$

Derivační složka umožňuje tedy předvídat budoucí chování procesu a využívat tuto znalost pro řízení. Z předchozí úvahy však vyplývají též omezení pro využití derivační složky a to v

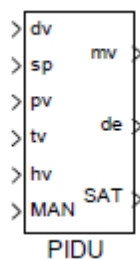
případech, kdy řízený systém obsahuje velké dopravní zpoždění, nebo kdy je regulovaná veličina zatížena velkým šumem měření[5].

Nastavování či seřizování parametrů PID regulátoru je snad nejdůležitější problém průmyslové regulace. Většina regulátorů v průmyslu (odhaduje se, že až 70% [10]) není vhodně nastavena. Cílem této kapitoly je stručně popsat alespoň hlavní přístupy užívané v současných komerčních regulátorech a programových systémech pro automatické seřizování PID regulátorů[5].

5.1 Metody nastavování regulátorů

- **Metoda pokus-omyl** - je nejčastějším postupem v praxi spočívajícím v přímém experimentování s uzavřenou smyčkou. Metodou pokus-omyl jsou voleny hodnoty parametrů regulátoru a podle tvaru odezvy na skok v požadované hodnotě (nebo uměle zavedený skok v poruše) se subjektivně usuzuje na jejich vhodnost. Existuje celá řada pravidel, které mají zefektivnit tento proces.
- **Zieglerovy-Nicholsovy metody** - V roce 1942 publikovali Ziegler a Nichols dvě přibližné metody pro nastavování regulátorů P, PI a PID [14]. Jimi empiricky získané vztahy překvapivě přežily až do současnosti a jsou v původní podobě nebo s různými modifikacemi stále používány jak při ručním, tak automatickém nastavování.

U soustavy kuličky na tyči kvůli astatismu 2.řádu nebudou standardní metody návrhu dávat akceptovatelný výsledek. Na tomto základě byl v regulačním algoritmu použit blok PIDU ze knihovny bloků v prostředí RexDraw. Do tohoto bloku, který nahradí standardní regulátor v regulační smyčce, jsou dosazeny proporcionální, integrační a derivační konstanty. Tyto konstanty jsou vygenerovány prostřednictvím algoritmu nástroje pidtool (pidtune) v prostředí aplikace Matlab R2010b.



Obr. 12 Blok regulátoru PIDU

5.2 Návrh regulátoru prostřednictvím pidtune, pidtool

V programovatelném prostředí aplikace Matlab R2010b byly pomocí nástroje pidtune vygenerovány regulační konstanty následovně:

- Nejprve do hlavního příkazového okna Command Window je zadán přenos soustavy (v tomto případě se jedná o přenos modelu odvozený v kapitole 2.2) a vložen do proměnné G .

```
>> G=tf([7.1428],[1 0 0])
```

Transfer function:

7.143

s^2

- Následuje příkaz pro použití nástroje pidtune, který vygeneruje regulační konstanty.

```
>> [C Info] = pidtune(G,'pid')
```

Continuous-time PID controller in parallel form:

$$\frac{1}{K_p + K_i * \frac{1}{s} + K_d * s}$$

with $K_p = 0.070001$, $K_i = 0.0011876$, $K_d = 0.12243$

Info =

Stable: 1

CrossoverFrequency: 1

PhaseMargin: 60.0000

Jak lze pozorovat z výpisu Command Window, nástroj pidtune vygeneroval následující regulační konstanty. Proporcionální **$K_p = 0.070001$** , integrační **$K_i = 0.0011876$** a derivační **$K_d = 0.12243$** .

Dalším nástrojem pro návrh regulačních konstant je v prostředí aplikace Matlab pidtool. Pro ověření a porovnání výsledků metod byl tento nástroj použit následovně. V programovacím a simulačním prostředí Matlab R2010b byl, podobně jako u předchozího nástroje, zadán přenos soustavy (resp. modelu kuličky na tyči) a vložen do proměnné G .

```
>> G=tf([7.1428],[1 0 0])
```

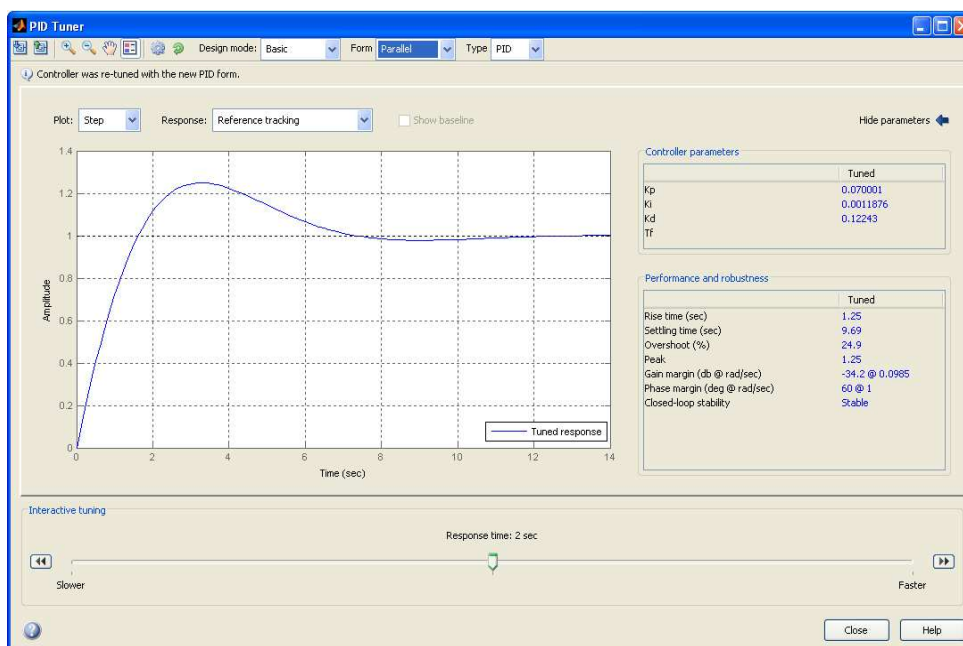
Transfer function:

7.143

s^2

Po výpisu přenosu v hlavním okně Command Window je zadán příkaz pro spuštění nástroje pidtool. Nyní je otevřeno nové okno nástroje pidtool (viz obr. 13).

```
>> pidtool(G)
```



Obr.13 Ukázka nástroje pidtool

Jak lze vidět na obrázku 13, v pravé části se nacházejí parametry a regulační konstanty vygenerované nástrojem, které se shodují s regulačními konstanty nástroje pidtune. Výhodou tohoto nástroje je, že uživatel má možnost dále upravovat jednotlivé vlastnosti regulace, např. dobu ustálení. Další výhodou je volba tvaru generovaných regulačních konstant apod.

Blok PIDU je základní blok pro vytvoření úplného regulátoru PID (P, I, PI, PD, PID, PI+S). V nejjednodušším případě může pracovat zcela samostatně a plnit standardní funkci PID regulátoru s dvěma stupni, volnosti v automatickém (MAN = 0) nebo manuálním režimu (MAN = 1)[5].

V automatickém režimu v lineární oblasti realizuje zákon řízení daný vztahem

$$U = \pm K \left[bW - Y + \frac{1}{T_i s} E + \frac{T_d s}{T_d s / N + 1} (cW - Y) \right] + Z \quad (32)$$

kde $U(s)$ je obraz akční veličiny mv, $W(s)$ je obraz požadované hodnoty sp, $Y(s)$ je obraz regulované veličiny pv, $Z(s)$ je obraz dopředné vazby dv a K , T_i , T_d , N , b , c jsou parametry regulátoru. Znaménko pravé strany je definováno parametrem RACT, určujícím směr působení akční veličiny mv na regulovanou veličinu pv (RACT=0: větší mv ® větší pv; RACT=1: větší mv ® menší pv). Rozsah výstupu mv je omezen saturačními mezemi lolim a hilim. Propojením výstupu mv se vstupem tv a vhodnou volbou parametru tt dosáhneme žádaného chování regulátoru při dosažení saturačních hodnot mv[5].

Jelikož nástroj pidtune generuje regulační konstanty v paralelním tvaru, je důležité tyto konstanty převést do ISA standardního tvaru (32). To z důvodu, že do regulačního bloku PIDU je nutno tyto konstanty zadávat pouze ve standardním tvaru ISA.

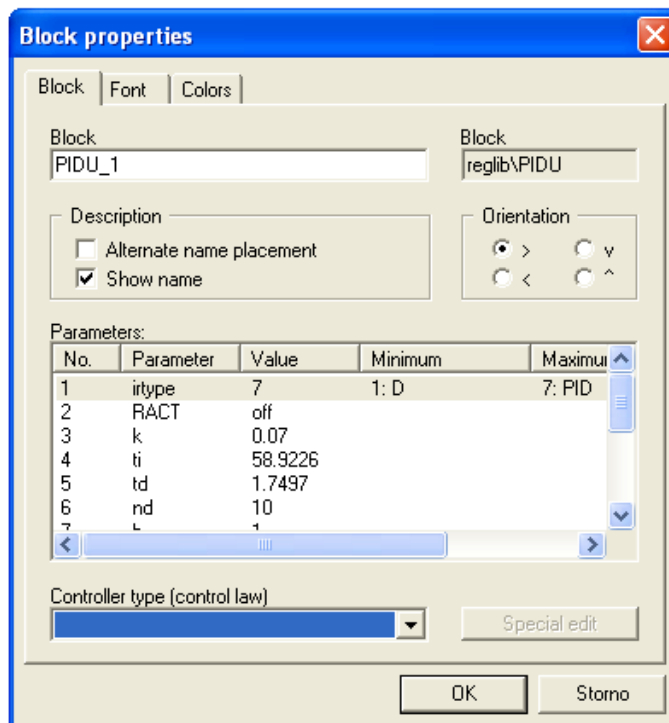
Vztah pro přepočítání parametrů mezi standardním a paralelním tvarem

$$U = K''(bW + Y) + \frac{K_i''}{s} E + \frac{K_d'' s}{\frac{K_d''}{NK''} s + 1} (cW - Y) \quad (33)$$

Přepočítání z paralelního tvaru na standardní ISA tvar

$$\begin{aligned} 0,07 + \frac{0,001188}{s} + 0,12248s &= 0,07 \left(1 + \frac{0,001188}{0,07 s} + \frac{0,12248}{0,07} s \right) = \\ &= 0,07 \left(1 + \frac{1}{\frac{0,07}{0,001188} s} + \frac{0,12248}{0,07} s \right) = 0,07 \left(1 + \frac{1}{58,9226s} + 1,7497s \right) \end{aligned}$$

Jak vyplývá z přepočtu na standardní tvar (32), regulační konstanty byly modifikovány na hodnoty proporcionální **Kp = 0,07**, integrační **Ki = 58,9226** a derivační **Kd = 1,7497**.



Obr. 14 Definitivní nastavení regulačního bloku PIDU

Definitivní nastavení regulačního bloku PIDU je následující. Parametr irtype je nastaven na hodnotu 7, což určuje typ regulace PID. Druhý parametr RACT (převrácení vstupní hodnoty) není aktivní. Parametry 3,4 a 5 jsou regulační konstanty, které jsou nastaveny na výše zmíněné hodnoty z ISA tvaru (viz obr. 13). Šestý parametr nd je filtr derivační složky nastavený implicitně na hodnotě 10. Následujícím nastavovaným parametrem je tt, kterým volí časová konstanta vyhledávání.

Doporučená hodnota tt je určena vztahem

$$tt \leq \sqrt{T_i T_d} \quad (34)$$

Následující parametry 10 a 11 jsou hilim a lolim. Ty určují meze limitace na výstupu regulačního bloku PIDU. Jsou nastaveny dle kapitoly o linearizaci pracovního bodu 4.1 na hodnotu 0,01 a hodnotu -0,01. Posledním nastavovaným parametrem je icotype, kterým určujeme typ výstupu regulačního bloku na analogový.

5.3 Návrh regulátoru metodou PID

Otevřený smyčkový přenos pro model kuličky na tyči je

$$\frac{R(s)}{\Theta(s)} = -\frac{mgd}{L\left(\frac{J}{R^2} + m\right)} \cdot \frac{1}{s^2} \quad (35)$$

Požadavky pro regulaci jsou následující:

- doba ustálení je méně než 3 sekundy
- překmit menší než 5%

Nejprve je sledována odezva systému při používání proporcionálního systému. Poté je v případě potřeby doplněna derivační a integrační složka.

Přenos uzavřené smyčky pro proporcionální regulování má hodnotu proporcionálního zesílení (kp) až 100.

Toto je modelováno v prostředí Matlab R2010b následovně

```
m = 0.111;  
R = 0.015;  
g = -9.8;  
L = 1.0;  
d = 0.03;  
J = 9.99e-6;  
K = (m*g*d) / (L*(J/R^2+m)); %simplifies input  
num = [-K];  
den = [1 0 0];  
kp = 1;  
numP = kp*num;  
  
[numc, denc] = cloop(numP, den)
```

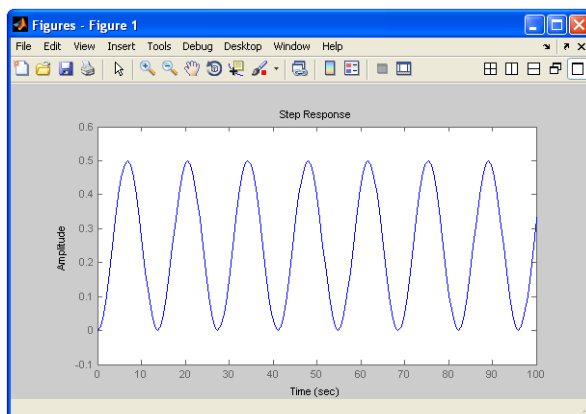
Výsledný čítec a jmenovatel je

```
numc =  
0 0 0.2100  
denc =  
1.0000 0 0.2100
```

Nyní následujícím příkazem bude model systému reagovat na vstupní skokovou odezvu ve výši 0,25m.

```
step(0.25*numc, denc)
```

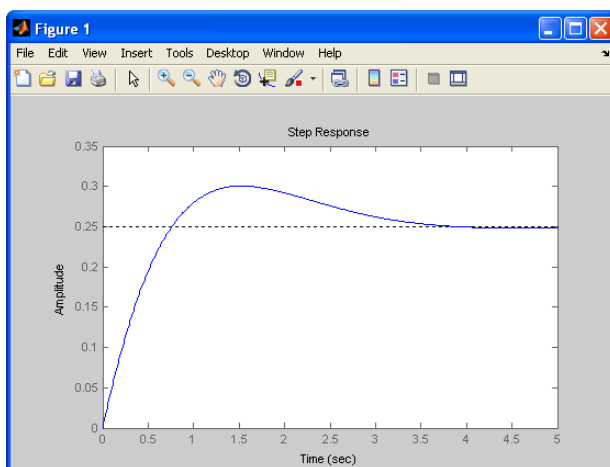
Po aplikaci příkazu je zobrazena tato charakteristika



Obr. 15 Výstupní odezva systému na vstupní skok

Jak lze vidět na obrázku 15, že zesílením proporcionálního zesílení se nestává systém stabilním. Je odzkoušeno, že při změně hodnoty k_p je systém nestabilní. Z tohoto důvodu je zapotřebí přidat do regulátoru derivační složku. Následujícím algoritmem se spustí a zobrazí odezva systému na regulaci s derivační složkou (kd).

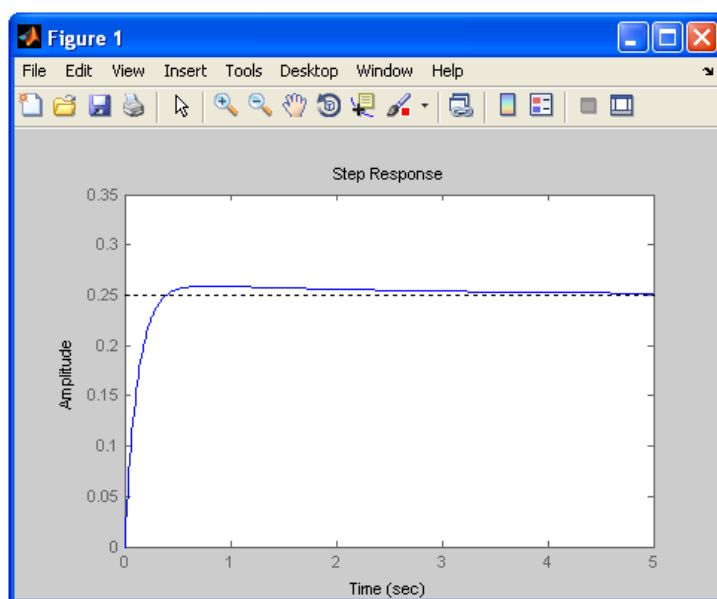
```
m = 0.111;
R = 0.015;
g = -9.8;
L = 1.0;
d = 0.03;
J = 9.99e-6;
K = (m*g*d)/(L*(J/R^2+m)); %simplifies input
num = [-K];
den = [1 0 0];
kp = 10;
kd = 10;
numPD = [kd kp];
numh = conv(num, numPD);
[numc, denc] = cloop(numh, den);
t=0:0.01:5;
step(0.25*numc,denc,t)
```



Obr. 16 Výstupní odezva systému s derivační složkou

Na obrázku 16 je potvrzeno, že se systém vlivem přidané derivační složky stává stabilní. Rovněž lze z obrázku 16 pozorovat, že parametry překmit a doba ustálení jsou na kladené požadavky příliš vysoké a je třeba tyto hodnoty snížit. Z tutoriálu stránky PID v sekci o vlastnostech P, I a D je zřejmé, že zvýší-li se derivační složka k_d , sníží se překmit a tím i doba ustálení. Ke snížení překmitu, může proporcionální složka k_p zvýšit mírně dobu náběhu.

Experimentálním dosazováním bylo zjištěno, že pro dosažení kritéria jsou ideální tyto hodnoty $k_p = 15$ a $k_d = 40$ (viz obr. 17).



Obr. 17 PD regulace systému

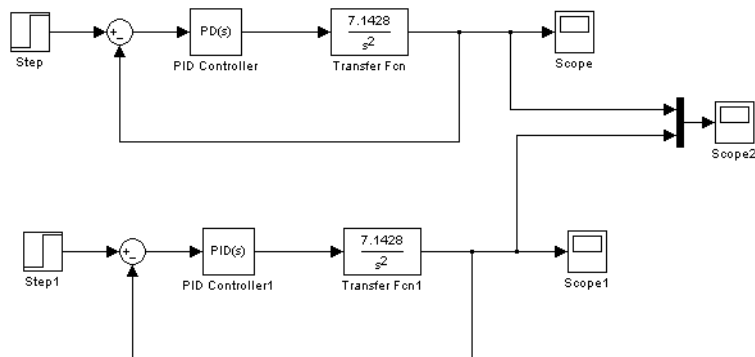
Jak vyplývá z obrázku 17, byly splněny počáteční kritéria a to i bez použití integrační složky. Doba ustálení je dosažena a odezva je menší, než dvě procenta.

5.4 Porovnání navržených regulátorů

Tato kapitola se zabývá porovnáním chování jednotlivých regulátorů v regulačních smyčkách v odezvě na jednotkový skok.

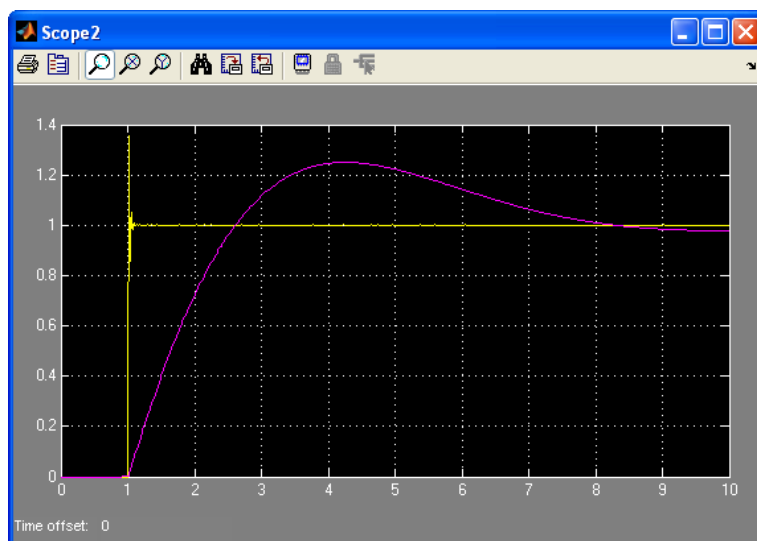
V simulačním prostředí aplikace Matlab Simulink byly vytvořeny jednoduché regulační smyčky s přenosem soustavy simulovaným funkcí Transfer Fnc a regulátorem PID, PD. Požadovaná hodnota je simulována jednotkovým skokem, který nabírá hodnoty od 0 do 1 v čase 1s. Výstup regulační smyčky je pozorován prostřednictvím bloku Scope v časovém intervalu 0 – 10s.

V simulačním prostředí Matlab Simulink byly vytvořeny dvě regulační smyčky. Každá z nich je regulována jedním z navržených regulátorů (viz obrázek 18).



Obr. 18 Porovnání regulátorů

Výsledná odezva na jednotkový skok regulačního systému je pozorována na výstupu regulační smyčky, kde se nachází reálná hodnota regulované amplitudy (y).

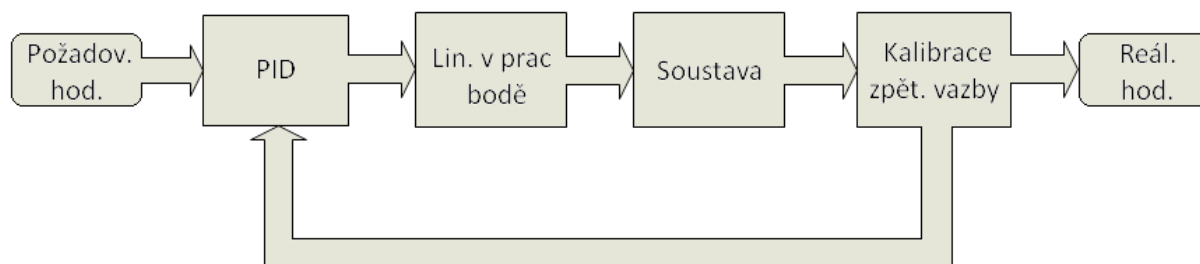


Obr. 19 Regulační děje navržených regulátorů

Na obrázku 19 jsou zobrazeny regulační děje obou navržených regulátorů. Žlutá křivka představuje regulaci navrženého PD regulátoru metodou PID a fialová křivka představuje regulační děj navrženého PID regulátoru prostřednictvím nástrojů pidtune a pidtool ve vývojovém prostředí Matlab R2010b. U regulace PD regulátorem je zjevná rychlost regulace a mnoho velkých překmitů v zanedbatelné době ustálení. U PID regulátoru má regulace ve srovnání s PD regulátorem dlouhou dobu náběhu a ustálení. Její velkou výhodou je plynulý přechod s menším překmitem, což je vzhledem k povaze problematiky kuličky na tyči nejvhodnější řešení regulace.

6 Návrh a realizace regulačního algoritmu

Vzhledem k tomu, že vývojové prostředí RexDraw umožňuje navrhovat algoritmus ve funkčních schématech, vychází tento algoritmus ze schématu regulační smyčky (viz obr. 20). Schémata jsou tvořena bloky z rozsáhlé knihovny systému REX, která je podrobně popsána v referenční příručce [5]. Výsledkem je soubor s příponou .mdl. Všechny bloky uvedené knihovny pracují v diskrétním čase, jejich velká část je automaticky diskretizována pro danou periodu vzorkování[5].



Obr. 20 Vývojový diagram regulačního algoritmu

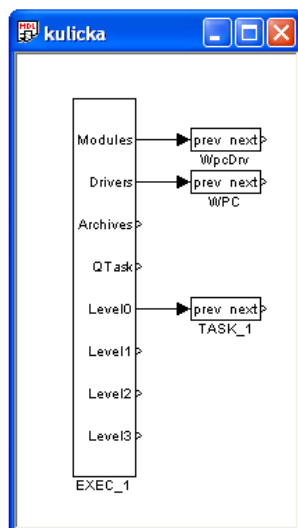
Vstupem do algoritmu regulační smyčky je hodnota požadované vzdálenosti kuličky od počátku ramene. Tato hodnota je následně importována do regulačního bloku PID, který je podrobně popsán v podkapitole 5.2. Výstupem regulačního bloku je akční veličina, neboli úhel náklonu ramene α . Ta je v následujícím bloku Linearizace v pracovním bodě přepočítána a realizovaná hodnotou napětí v dosahu 0 – 10V na analogovém výstupu programovatelného automatu. Tento výstup a následně i analogový napěťový vstup automatu je v regulačním algoritmu reprezentován blokem Soustava. Vstup, resp. výstup modelu vyjadřuje hodnotou napětí 10 – 0V reálnou vzdálenost kuličky. Pro dobrou činnost regulace je tato hodnota v napěťovém rozsahu přepočítána a kalibrován. Tuto činnost provádí blok Kalibrace zpětné vazby. Výstupem tohoto bloku je reálná hodnota, neboli aktuální pozice kuličky a také záporná zpětná vazba pro blok PID.

Jak je výše zmíněno, PID blok je detailně popsán v podkapitole 5.2. Bloky Linearizace v pracovním bodě a Kalibrace zpětné vazby (výstupu regulační smyčky) jsou popsány a implementovány do blokového schématu v kapitole 4 Linearizace v pracovním bodě.

Vychází-li se z faktů, že všechny bloky vývojového diagramu regulátoru jsou v předchozích kapitolách popsány a navrženy, následuje realizace regulačního algoritmu.

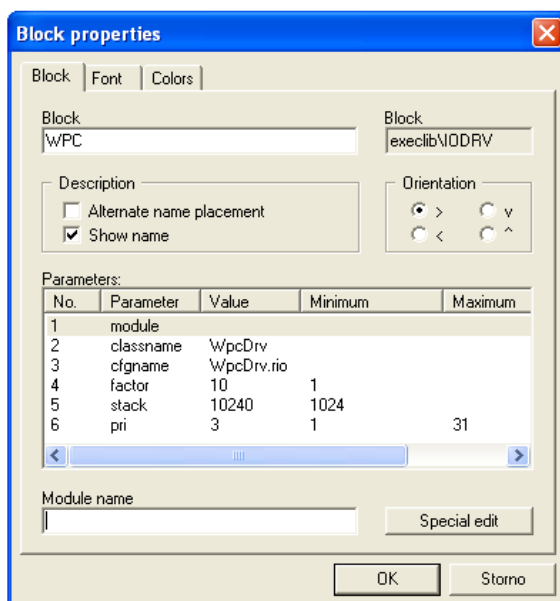
Algoritmus je realizován ve vývojovém prostředí RexDraw, které je součástí řídicího systému REX. Důležitou a nedílnou součástí pro chod regulace je soubor WpcDrv.rio, který slouží jako ovladač řídicího systému REX a je určen pro ovládání vstupů a výstupů jednotlivých modulů. Tento soubor je nutno vložit do složky, kde jsou uloženy i soubory .mdl s implementovaným algoritmem.

Pro realizaci regulačního algoritmu je nutno ve vývojovém prostředí RexDraw přidat ovladač do hlavního souboru. Z knihovny bloků RexDraw je použit hlavní blok programu EXEC_1 v okně vývojového prostředí pojmenovaném kulička.



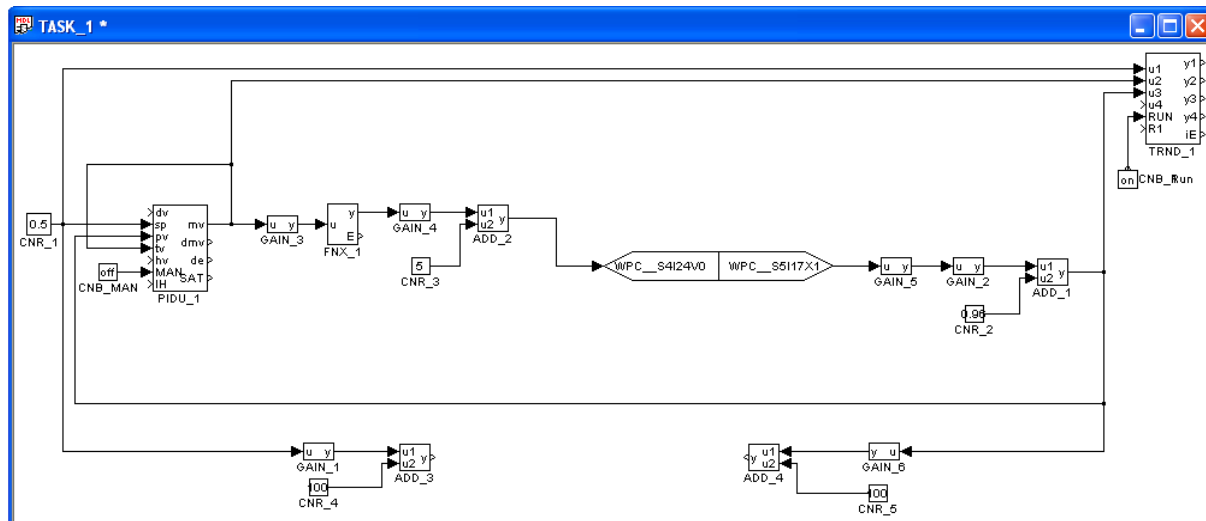
Obr. 21 Zařazení ovladače do projektu aplikace

Pro zařazení ovladače do projektu slouží dva zvýrazněné bloky. Nejprve je na výstup Modules exekutivy EXEC připojen blok s názvem WpcDrv typu MODULE, který nemá žádné další parametry[5]. V druhém bloku WPC typu IODRV, připojeném na výstup Drivers exekutivy jsou zadány následující parametry (viz obr. 22).



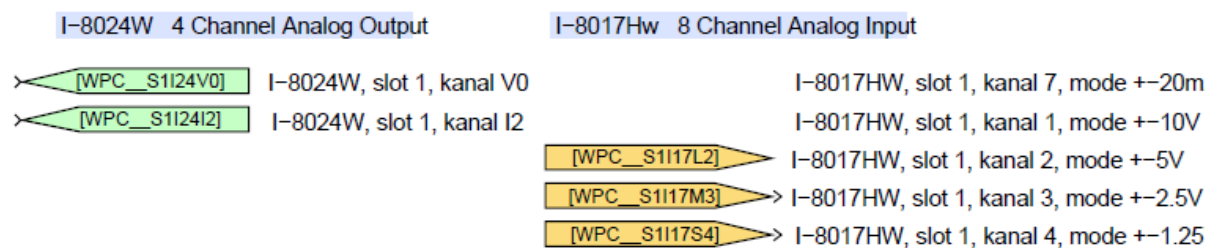
Obr. 22 Nastavení bloku WPC

Na obrázku 21 je vidět, že je připojený blok TASK_1 k hlavnímu bloku programu EXEC_1. Tento blok reprezentuje uživatelský program. V okně TASK_1 vývojového prostředí je tedy prostor pro tvorbu programu.



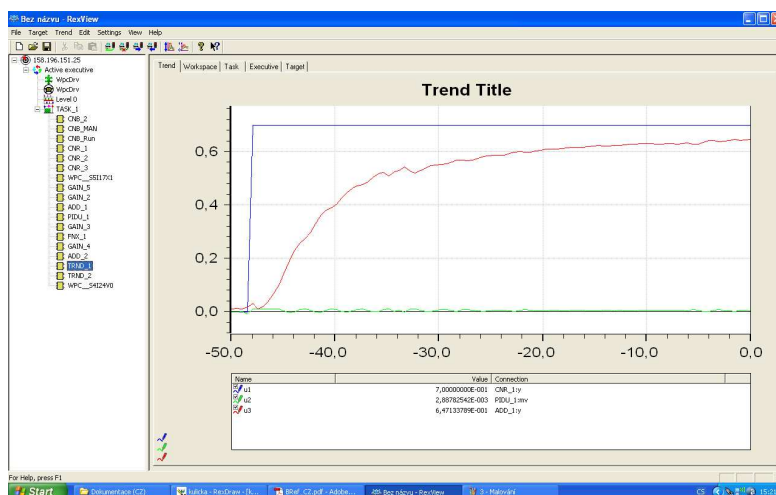
Obr. 23 Implementace regulačního algoritmu

Tzv. zdrojový kód navrženého regulátoru je zobrazen na obrázku 23 v okně TASK_1. V levé části obrázku se nachází blok CNR_1, což je matematický mlok pro zadávání konstanty. V tomto případě se jedná o nastavování požadované hodnoty. Napravo od CNR_1 je blok regulátoru PIDU_1, jenž je podrobně popsán v kapitole 5.2. Tento blok je zapojen dle předlohy v technické dokumentaci výrobce [6]. Na výstup tohoto navazuje skupina bloků GAIN_3, FNX_, GAIN_4, RNR_3 a ADD_2, což jsou bloky pro přepočet a linearizace v pracovním bodě podrobně popsané v kapitole 4.1. Dle vývojového diagramu následují bloky reprezentující soustavu. Jak bylo výše zmíněno soustavu ve zdrojovém kódu zastupují vlajky (odkazy) analogového a napětového vstupu a výstupu programovatelného automatu. Tyto vstupy jsou připojeny ke stupu a výstupu modelu kuličky na tyči. Možnosti nastavování vlajek jsou zobrazeny níže na obrázku 24.



Obr. 24 Možnosti nastavení vlajek

Z výstupu soustavy (resp. do vstupu regulátoru) je signál o pozici kuličky přiveden do skupiny bloků, které toto napětí přepočítávají a kalibrují pro zápornou zpětnou vazbu regulátoru a zároveň je tento signál přiveden do bloku TREND_1 který slouží k zobrazení všech přivedených hodnot (požadovaná hodnota, akční veličina a reálná hodnota) do grafu v diagnostickém prostředí RexView v závislosti na čase. Ukázka zobrazení hodnot v RexView prostřednictvím bloku TREND_1 je zobrazen níže na obrázku 25.



Obr. 25 Ukázka regulace v RexView

V dolní části zdrojového kódu se nachází na levé straně bloky pro přepočet požadované hodnoty pro import do vizualizačního prostředí a na pravé straně bloky pro přepočet reálné hodnoty rovněž pro import do vizualizačního prostředí.

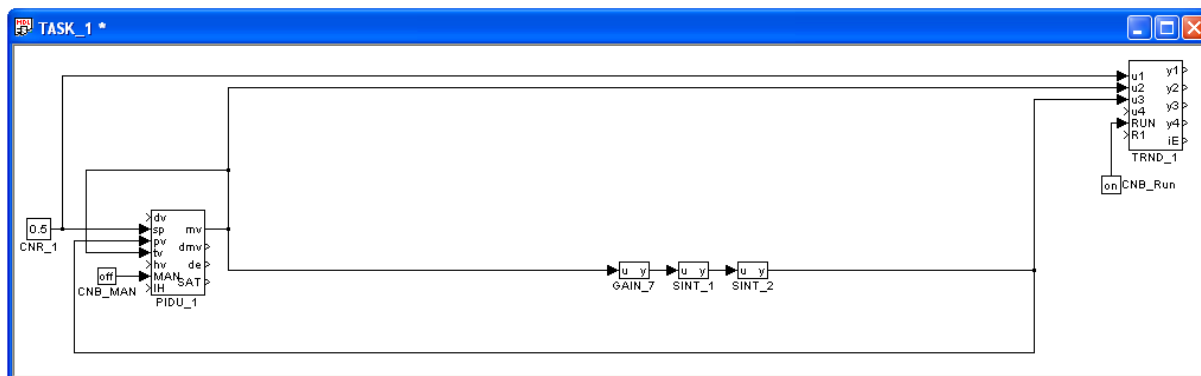
6.1 Simulace realizovaného regulátoru

Simulace regulátoru, resp. vygenerovaných regulačních konstant byla vytvořena již výše (v podkapitole 5.4, obrázek 19) a to v simulačním prostředí aplikace Matlab Simulink pro porovnání dvou navržených regulátorů PD a PID.

Z důvodů reálnějšího výsledku (na rozdíl od teoretického v prostředí Simulink) je zvolena jiná metoda simulace navrženého regulátoru.

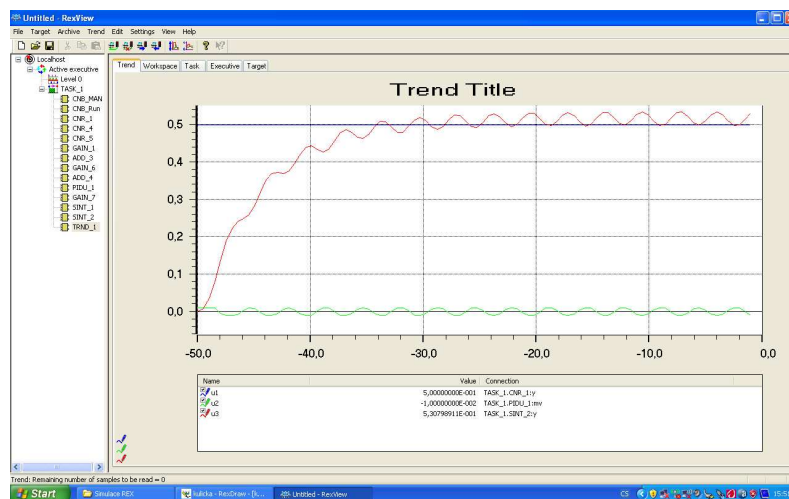
Ve vývojovém prostředí RexDraw je zdrojový kód upraven. Vlajky vstupu a výstupu programovatelného automatu byly nahrazeny blokem GAIN_7 se zesílením 7,1428 a dvou sériově navazujících bliků SINT_1, SINT_2 které slouží k integraci. Tyto bloky simulují přenos soustavy a tím nahrazují fyzikální model kuličky na tyči.

Dále byly odstraněny bloky pro přepočty, linearizaci v pracovním bodě a kalibraci které v činnosti simulace neplní žádný význam. Význam ztrácely i bloky WPC a WpcDrv připojené k hlavnímu bloku programu EXEC_1. Při ponechání těchto bloků nelze simulaci uskutečnit!



Obr. 26 Úprava zdrojového kódu pro simulaci

Jelikož simulace probíhá pouze v PC, je nutné změnit nastavení Target device hlavního bloku programu EXEC_1 z Windows CE na PC – Windows. Poté je nezbytně důležité spustit program RexCore nacházející se v složce bin nainstalovaného REX Controls. Při nahrávání upraveného programu je potřeba místo IP adresy programovatelného automatu zadat Localhost. Výsledek simulované regulace lze stejně jako u reálné regulace pozorovat v diagnostickém prostředí RexView.



Obr. 27 Simulace regulace

Na obrázku 27 je zobrazen simulovaný regulační děj. V porovnání s teoretickou simulací v prostředí Simulink má simulace v řídicím systému REX odlišný průběh. Vzhledem k tomu, že byly regulační konstanty generovány ověřeným nástrojem a vzhledem k astaticismu druhého řádu modelu, naplňuje nežádoucí zvlnění regulační křivky teoretický předpoklad.

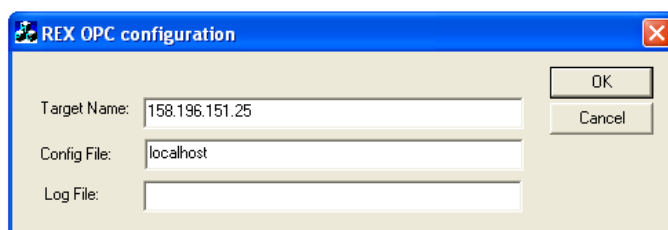
7 Graficko – uživatelského rozhraní

Vizualizaci lze realizovat mnoha způsoby. Rozdíly mezi nimi jsou v použitém vývojovém vizualizačním prostředí. To mohou být například Excel, In Touch, Genesis 32 a mnoho jiných.

V této práci je vizualizace realizována prostřednictvím vývojového prostředí Promotic. Jeho hlavní výhodou, například od vývojového prostředí In Touch, jsou možnosti použitelných prvků z knihovny Promoticu.

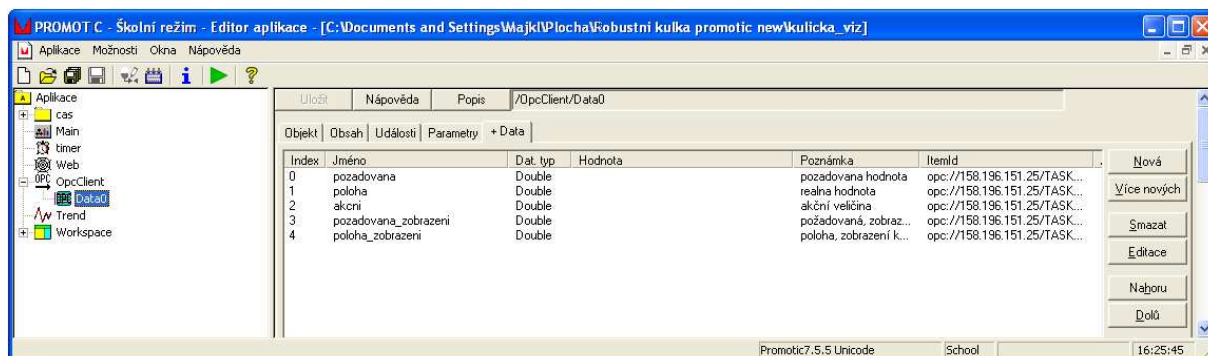
Promotic je prostředí, které patří do skupiny prostředí tzv. OPC klientů. Tzn. že jeho komunikaci zprostředkovává OPC server přes komunikační rozhraní standard TCP/IP, což má z hlediska uživatele nesmírné výhody. Hlavní výhodou je masivní rozšíření Ethernetové komunikace pomocí TCP/IP protokolu čtvrté generace. Toto rozhraní dále předá informace hlavnímu řídicímu programu RexCore, který řídí technologický proces, resp. vstup/výstup programovatelného automatu.

Vše je popsáno v kapitole 3.2 a zobrazeno na obrázku 4.



Obr. 28 Ukázka nastavení OPC serveru

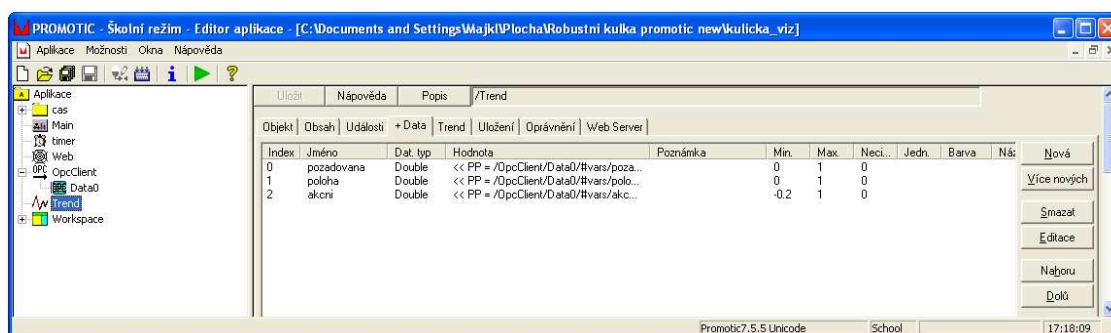
Pro realizaci vizualizace GUI (graficko-uživatelského rozhraní) je nejprve třeba nastavit OPC server (viz obrázek 28). V kolonce Target Name je vyplněná IP adresa řídicího automatu. V kolonce Config File je vyplněn localhost. Ostatní nastavení zůstává nezměněno.



Obr. 29 Nastavení proměnných v Promoticu

Na obrázku 29 je zobrazeno nastavení importu proměnných prostřednictvím OpcClientu. Jedná se o proměnné, které budou zobrazeny ve grafu vizualizace jako např. akční veličina, požadovaná hodnota a skutečná poloha kuličky. Dále pak proměnné které jak bylo zmíněno v předchozí kapitole slouží jako souřadnice pro dynamické zobrazování polohy kuličky v animaci modelu. Jedná se o pozadovana_zobrazeni a poloha_zobrazeni.

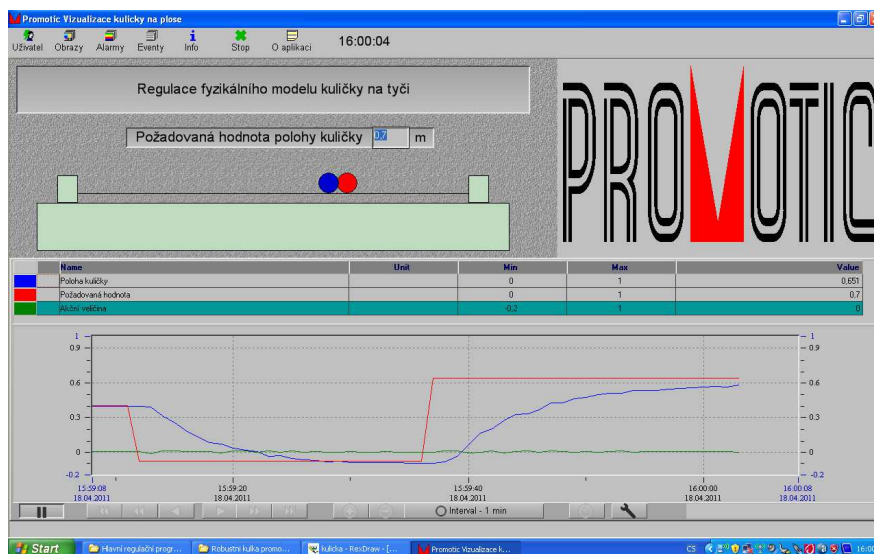
Dále lze z obrázku pozorovat i jednotlivě vytvořené prvky aplikace jako jsou např. Timer, který slouží jako generátor reálného času zobrazování průběhu regulace v grafu. Zobrazovací graf zajišťuje Trend, jehož nastavení proměnných je zobrazeno na obrázku 30.



Obr. 30 Nastavení proměnných v grafu

Graf zobrazuje podobně jako u diagnostického prostředí RexView požadovanou, akční a reálnou hodnotu.

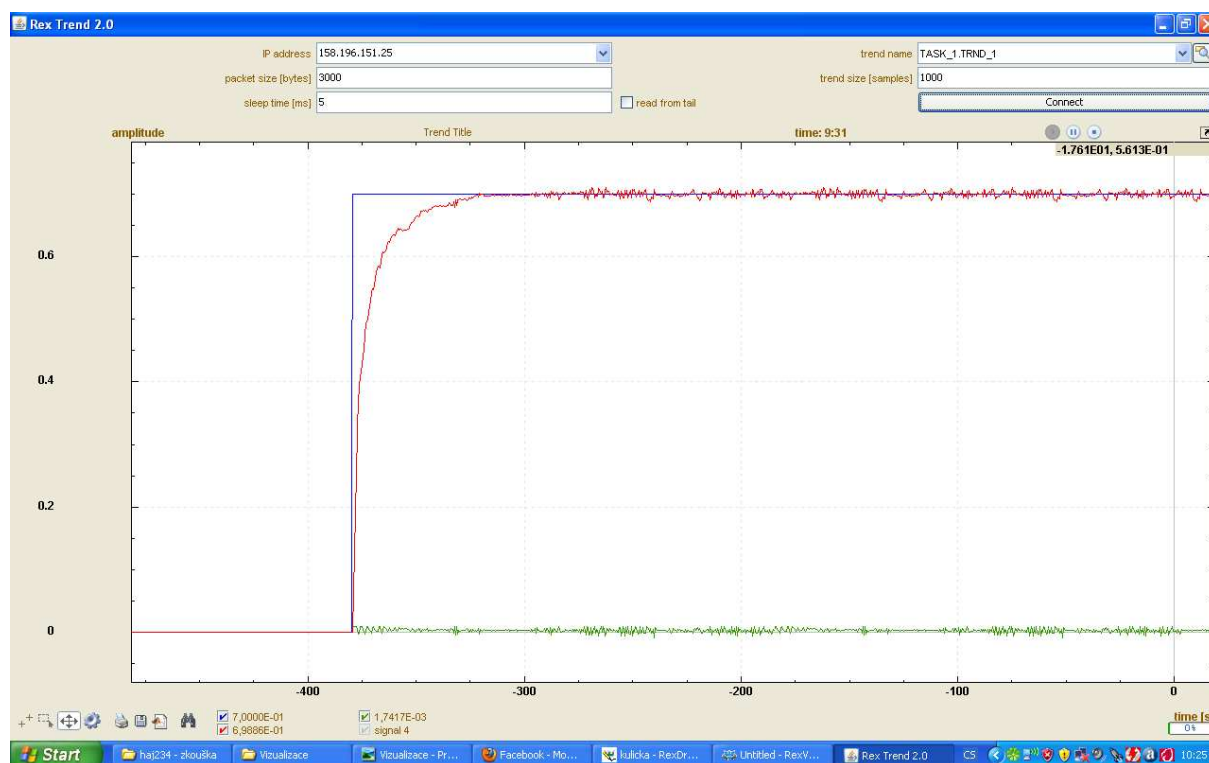
Hlavní položka Main, jejíž záložka obraz a tlačítko Editace grafického obsahu je z hlediska tvorby vizualizace nejdůležitějším členem aplikace. Tento člen poskytuje prostor pro tvorbu hlavní vizualizační stránky. Na obrázku 31 je finální verze vizualizace pro regulaci modelu kuličky na tyči.



Obr. 31 Vizualizace regulace kuličky na tyči

Vedle vizualizačního prostředí vytvořeného ve vývojovém prostředí Promotic je regulační děj pozorován i v Rex Trend 2.0, což je vysoce přesný graf poskytovaný výrobcem Rex Controls. Tento trend je vytvořený na Java platformě a volně přístupný na webových stránkách <http://www.rexcontrols.com>. Pozn. k použití Rex Trend je třeba nainstalovat Java knihovny do PC.

Velké výhody tohoto trendu jsou např. jednoduché připojení k WinPAC a následné zobrazení proměnných z bloku Trend_1 regulačního algoritmu. Další výhodou je sleep time, který je implicitně nastavená na domu 5 ms, což je dostačující vzorkovací perioda pro přesně zobrazující výsledek regulace.



Obr. 32 Ukázka regulace v Rex Trend 2.0

Na obrázku 32 je zobrazen regulační děj modelu kuličky na tyči prostřednictvím vizualizačního prostředí Rex Trend 2.0. V horní části obrázku jsou umístěny textboxy a scrollboxy pro nastavování jednotlivých parametrů a připojení. Uživatel tohoto trendu po nastavení IP adresy daného programovatelného automatu pak dále nastavuje dobu zobrazení, vzorkovací periodu, počet datových packetů a podobně jako ve RexView i blok programu, ze kterého čerpá proměnné.

Zobrazený průběh regulace se v pozici požadované hodnoty mírně rozkmitává. To je způsobeno velice přesným měřením napětí generované čidlem polohy kuličky. Skutečná pozice kuličky je tedy ustálená na požadované hodnotě resp. na požadované vzdálenosti od počátku ramene.

8 Zhodnocení, závěr

V bakalářské práci jsou splněny všechny zásady pro vypracování dle zadání. V jednotlivých kapitolách je popsán model kuličky na tyči, řídicí systém REX včetně programovatelného automatu WinPAC-8000, návrh a realizace PID regulátoru, linearizace, implementace regulačního algoritmu a graficko-uživatelské rozhraní GUI.

V kapitole 2 je model popsán fyzikálně a matematicky. Fyzikální popis udává základní vlastnosti modelu a popisuje jeho mechanickou, elektrickou a programovou část. Matematický popis se zabývá, prostřednictvím pohybových rovnic, vlivem náklonu ramene na pohyb kuličky pro odvození přenosové funkce této soustavy. Odvozený přenos soustavy je klíčový pro návrh regulátoru.

Následující kapitola 3 se zabývá popisem WinPAC-8000 a řídicího systému REX. U programovatelného automatu WinPAC-8000 jsou popsány především jeho parametry, vlastnosti a možnosti využití. Kombinace tohoto automatu s operačním systémem Windows CE a řídicím systémem REX tvoří široce variabilní systém pro řízení a regulaci procesů v automatizaci.

Čtvrtá kapitola se zabývá přepočtem, kde na výstup regulátoru je úhel α (náklon ramene), na výstupní napětí automatu a jeho linearizaci v rozsahu 0 – 10V. Dále se zabývá přepočtem a kalibrací vstupního napětí do automatu (zpětná vazba regulátoru) na hodnotu skutečné vzdálenosti kuličky od počátku ramene. Tyto přepočty jsou modifikovány do blokových schématů, které jsou následovně použity v realizaci regulačního programu.

V kapitole 5 je navržen PID regulátor a porovnán s alternativním PD regulátorem. Prostřednictvím nástroje pidtool aplikace Matlab jsou vygenerovány regulační konstanty. Tyto konstanty jsou přepočítány na standart ISA tvar a aplikovány do regulačního bloku PIDU z knihovny vývojového prostředí RexDraw.

Tento regulační blok je v následující kapitole 6 použit v implementaci regulačního algoritmu. Realizovaný regulační systém je simulován v řídicím systému REX. Při porovnávání simulované regulace se skutečnou regulací modelu je zjevné, že potřebná doba ustálené přechodového děje je u skutečné regulace delší, což je způsobeno neideálním konstrukčním zpracováním modelu.

Pro nastavování požadované hodnoty regulace je vytvořeno GUI prostředí, kde lze pozorovat regulační děj v trendu, který zobrazuje křivku akční veličiny, požadovanou a skutečnou pozici kuličky. Dále lze v GUI pozorovat i animaci regulace modelu kuličky na tyči kde jedna kulička znázorňuje požadovanou a druhá reálnou polohu kuličky.

Literatura

- [1] NOSKIEVIČ, P. *Modelování a identifikace systémů*. 1. vyd. Ostrava: MONTNEX, a. s., 1999. 276 s. ISBN 80-7225-030-2.
- [2] VAVŘÍN, P. *Teorie automatického řízení 1 (Lineární spojité a diskrétní systémy)*. 2. přeprac. vyd. Brno: VUT Brno, 1991. 158 s. ISBN 80-214-0244-X.
- [3] ŠULC, B. *Teorie automatického řízení s počítačovou podporou*. 1. vyd. Praha: ČVUT, 1999. 154 s. ISBN 80-01-02044-4.
- [4] Firemní dokumentace Matlab and Simulink (The MathWorks, Inc.)
- [5] Firemní dokumentace Rex Controls.
- [6] Firemní dokumentace ICP DAS k programovatelným automatům.
- [7] Technická dokumentace Fyzikální model kuličky na tyči.
- [8] VOJČINÁK, Petr, PIEŠ, Martin, HÁJOVSKÝ, Radovan. Návrh LQG řízení pro fyzikální model kuličky na tyči. In Humusoft, s.r.o.. *Technical Computing Prague 2009 : Sborník příspěvků 17.ročníku konference*. Praha : Humusoft, s.r.o, 2009. s. 109. 1x CD-ROM. Kongresové centrum ČVUT, Praha, November 14, 2009. Dostupný z WWW: <http://dsp.vscht.cz/konference_matlab/MATLAB09/prispevky/109_vojcinak.pdf>. ISBN 978-80-7080-733-0.